## BICT Course Evaluation

**Programming Practices**

Obviously, being a software development project, I made significant use of my knowledge from all previous software engineering courses, from SE101 to PR301, although interestingly not as much as might be expected. Since this project was focussed on web development, it didn’t utilise so much of the content which was largely focussed more on object oriented programming. Regardless, these courses still provided the fundamentals and the more advanced nuances of programming and following best practices that were necessary for the success of this project. Principles like DRY (Don’t repeat yourself) and SoC (Separation of concerns) all came into play for this project.

**MVC / Design Patterns**

The design and structure of the software for this project was similar to the Model-View-Controller (MVC) design architecture, first introduced to me in PR282 and further used in PR283 and PR301. Having an understanding of MVC helped in drawing parallels with the architecture we were using, which provided an easier and deeper understanding of how to use the architecture and the benefits that using it provided. I was also able to relate the architecture to my knowledge of Design Patterns gained from PR301. The architecture we used was in some ways similar to the “Façade” design pattern in that it provided a higher level abstraction (the API) of another sub-system (the DAO) to make it easier to use.

**Agile Development**

Agile development played a major role in the success of this project. Using our knowledge of methodologies gained from IS201, we were able to develop our project following an agile approach. This meant we were able to handle new and changing requirements requested by the client throughout development, even when nearing completion. If we had followed a standard waterfall approach and stuck purely to our original requirements and ideas for the project, it very likely wouldn’t have turned out as good as it did and it may not have suited the client’s needs well enough in the end.

**Recommendations**

As previously mentioned, synchronicity in programming is not a concept that was really covered in any of the previous courses, or if it was it was mentioned only briefly, and yet it seems like a rather important concept. Especially since it can be difficult to grasp at first, I think it would be beneficial to include AJAX or some other form of asynchronous programming as a part of one of the software engineering courses, perhaps PR301 since it’s a slightly more advanced concept.

I would also suggest possibly covering the use of libraries slightly more. While it was touched on in the courses, there weren’t really any assignments or projects that made use (or allowed use) of libraries in a way that actually provided experience in using them. I think it would be useful to provide that experience, especially in using libraries commonly used in the industry, such as jQuery. I can understand the need for students to create most things themselves for the sake of learning, especially since if it was allowed a lot of projects would be far too easy using libraries, but I believe encouraging or maybe even requiring the use a library in at least one assignment would be a good thing for providing some experience that can benefit students when they move into the work place.